PhD research project presentation

"A generic framework for many-valued logics with a finite number of truth values by means of ordered functional complete truth tables"

koen.lefever@vub.ac.be

Centrum voor Logika en Wetenschapsfilosofie - CLWF VUB

January 16th, 2012

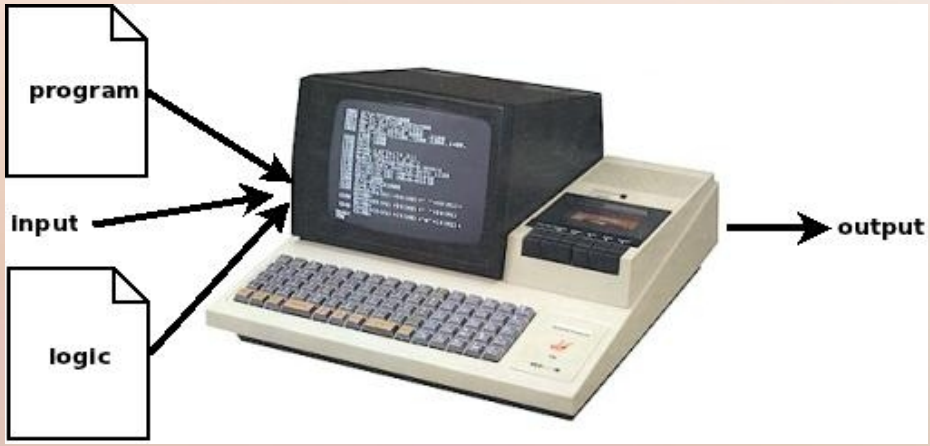
Vesalius College
Brussels

A generic framework for many-valued logics with a finite number of truth values by means of ordered functional complete truth tables

- **By Koen Lefever** (about me)
- **Purpose:**
  - To construct a way to describe and work with many-valued logics in a way which is very generic.
  - Study of multi-valued logics in automated inference.
- **Constraint:**
  - Finite number of truth values: a computer cannot process actual infinities.
- **Promotor:**
  - Prof. dr. Jean Paul van Bendegem, CLWF-VUB
- **Research Sponsor:**
  - Vesalius College Brussels

A generic framework for many-valued logics with a finite number of truth values by means of ordered functional complete truth tables

- **Philosophical backgrounds:**
  - The basic ideas of this approach were the result of another question: would it be possible to give a computer a description of any logic, and expect it to be able to run a given program by that logic.



  - Any possible logic is probably an impossible aim, so a more modest aim would be to determine what set of logics can be described in such a framework and the properties which these logics may need to have.
  - The set of finite multi-valued logics is a large and important subset of logics:
    - It includes uncertainty (e.g. Łukasiewicz & Kleene) and paraconsistent (e.g. Jaśkowski) logics.
  - A computer program contains more logic than just the logical operators. In other words: the “logic” and the “code” of a program may not be distinct. Therefor, it makes sense to implement the code which handles the logic as a library to be used by computer programs. A start at this is attempted here on a practical level by providing a multi-valued logic toolkit software library in Python. This may help to clarify on the differences and overlaps between “logic” and “code”. How do multi-valued logics behave in automated inference, differences & similarities between different multi-valued logics?

A generic framework for many-valued logics with a finite number of truth values by means of ordered functional complete truth tables

- **Methods:**
  - Use of combinatorial mathematics on multi-valued logics in the spirit of Boole and De Morgan.
  - Ordered functional complete truth tables will be used to generate and study various many-valued logics.
    - Functional complete truth tables have been described and studied before by Emil Post (1921), William Wernick (1942) and Donald Knuth (2008).
    - The difference with the functionally complete tables studied by the above authors, is that the tables in this proposal are ordered: n-valued logics are treated as number systems with base n. This provides for a straightforward way to number and order operators in n-valued logic and to fastly calculate (only a base conversion is needed) the truth values column for this operator in the ordered functional complete table.
    - A multivalued logic is defined by giving an ordered list of truth values and a list of designated truth values (both lists can be generated automatically or be constructed by the user in the computer implementation). Then columns of note of the Ordered functional complete truth tables can be defined as operators. The order of the truth values in the above list can be at random or can reflect a hierarchy in the truth values.

A generic framework for many-valued logics with a finite number of truth values by means of ordered functional complete truth tables

- **Methods (continued):**
  - Ampheck operators:
    - Ampheck (Greek, 'cutting both ways') operators are generalizations of the NOR/Peirce arrow and its dual the NAND/Sheffer strike to n-valued logics.
    - a method is proposed to determine n! ampheck operators. (seems to be basically the same as in Norman Martin (1976)).
    - Ampheck operators will be used to it faster and easier to prove that a set of operators is functionally complete for an n-valued logic, since it suffices to show that at least one ampheck operator can be generated from the set to ensure functional completeness.
  - A study of generating parts of the functional complete table which contain wanted columns and exclude unwanted columns will be undertaken (cf. The work by John Slaney on automated reasoning).

A generic framework for many-valued logics with a finite number of truth values by means of ordered functional complete truth tables

- Ordered functionally complete truth table for binary logic:

			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	P	Q	0	P&Q	~(P⊃Q)	P	~(Q⊃P)	Q	P⊕Q	P∨Q	P↓Q	P≡Q	~Q	Q⊃P	~P	P⊃Q	P|Q	1
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
2	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
3	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

- Numbers of rows, columns and ampheck operators:
  - 2 truth values: 4 rows, 16 columns, 2 amphecks (columns 8 and 14)
  - 3 truth values: 9 rows, 19683 columns, 6 amphecks
  - 4 truth values: 16 rows, 65536 columns, 24 amphecks
  - 5 truth values: 25 rows, 29802322387695312 5 columns, 120 amphecks
  - n truth values: n^2 rows, $n^{(n^2)}$ columns, n! ampheck operators

A generic framework for many-valued logics with a finite number of truth values by means of ordered functional complete truth tables

- Ordered functionally complete truth table for ternary logics:

			0	113	4049	6723	19094	19305	19337	19418	19519	19682
	P	Q	0	P&Q	P∨Q	First ampheck	Jaśkowski P⊃Q	~P	Kleene P⊃Q	Łukasiewicz P⊃Q	Last ampheck	2
0	0	0	0	0	0	1	2	2	2	2	2	2
1	0	1	0	0	1	0	2	2	2	2	2	2
2	0	2	0	0	2	0	2	2	2	2	2	2
3	1	0	0	0	1	0	0	1	1	1	2	2
4	1	1	0	1	1	2	1	1	1	2	0	2
5	1	2	0	1	2	0	2	1	2	2	2	2
6	2	0	0	0	2	0	0	0	0	0	2	2
7	2	1	0	1	2	0	1	0	1	1	2	2
8	2	2	0	2	2	0	2	0	2	2	1	2

0 = false, 1 = third state (undefined or paradoxical), 2 = true

For Łukasiewicz & Kleene, only 'true' is designated

For Jaśkowski, both “true” and “paradoxical” are designated

A generic framework for many-valued logics with a finite number of truth values by means of ordered functional complete truth tables

- **Implementation:**
  - This approach allows us to define a many-valued logic by giving 3 lists:
    - A list of truth values
    - A list of designated truth values (all others are non-designated)
    - A list of operators, defined by their number in the ordered functional complete truth table
  - A software library to generate logics from descriptions has been created using the Python programming language:

```
lucasiewicz = MVL_Toolkit.Logic(['false', 'undefined', 'true'], ['true'])

def lucasiewicz_AND(P, Q):
    return lucasiewicz.operator(113, P, Q)
def lucasiewicz_OR(P, Q):
    return lucasiewicz.operator(4049, P, Q)
def lucasiewicz_NOT(P):
    return lucasiewicz.operator(19305, P, P)
def lucasiewicz_IF_THEN(P, Q):
    return lucasiewicz.operator(19418, P, Q)

def kleene_IF_THEN(P, Q):
    return lucasiewicz.operator(19337, P, Q)
```

A generic framework for many-valued logics with a finite number of truth values by means of ordered functional complete truth tables

- **Implementation** (continued):
  - Another example:

```
# 101-valued logic with all values up from 95 being designated
# this could be interpreted as a discrete probabilistic logic
# with significance at P=0.05)
percentage = MVL_Toolkit.Logic(range(0,101),range(95,101))
```

  - The toolkit provides methods to use those logics in computer programs (commands are typed by the user after the >>> prompt):

```
>>> lucasiewicz.nr_of_columns
19683
>>> lucasiewicz.column_to_nr(['undefined','false','false','false','true',
                              'false','false','false','false'])
6723
>>> lucasiewicz.operator(19337,'false','true')
'true'
>>> lucasiewicz_IF_THEN('undefined','undefined')
'true'
>>> kleene_IF_THEN('undefined','undefined')
'undefined'
```

  - Fast detection of ampheck operators are the next thing to be implemented.
    - Operator 6723 in the above example is the first ampheck for ternary logic.

A generic framework for many-valued logics with a finite number of truth values by means of ordered functional complete truth tables

- **Already done:**
  - Basic lay-out of the framework, basic methodology of using ordered functional complete truth tables
  - Started the software component which has already some features (including a brute-force approach to determine ampheck operators) and supports several multi-valued logics (including Dunn/Belnap four-valued logic, Łukasiewicz, Kleene & Jaśkowski three-valued logics and a 101-valued discrete probabilistic logic).
  - Project website:
    - http://code.google.com/p/mvl-toolkit/
    - Introduction to the concepts of the framework, in which the earlier mentioned methods are explained
    - Examples on how to use the toolkit
    - Overview of logics which are (to be) implemented in the mvl toolkit:
    - Program code of the software library with Subversion version control

A generic framework for many-valued logics with a finite number of truth values by means of ordered functional complete truth tables

- **To do:**
  - 2012:
    - Reading and studying the state of the art on multi-valued logics (see bibliography for the most important starting points),
    - Completion of methodology, publication of first research paper,
    - Advancements in the software component (add other logics such as finite-valued Gödel logics and general finite-valued versions of the above mentioned three-valued logics, implement fast determination of n! ampheck functions, generating partial tables).
  - 2013:
    - Formal consolidation and consistency, proof of theorems,
    - Finishing the software component,
  - 2014:
    - Conclusions on multi-valued logics and specifically on mechanical inference in them,
    - Writing and editing,
  - 2015:
    - Completion & defending of the dissertation.
  - Continuation (or perhaps still within the time frame of the project if everything goes smoothly):
    - Building a multi-valued logic inference engine.

A generic framework for many-valued logics with a finite number of truth values by means of ordered functional complete truth tables

- **Bibliography:**

- Emil L. Post: **Introduction to a General Theory of Elementary Propositions**, American Journal of Mathematics, Vol. 43, No. 3, (July 1921), pp. 163-185
- William Wernick: **Complete Sets of Logical Functions**, Transactions of the American Mathematical Society, Vol. 51, No. 1. (Jan., 1942), pp. 117-132.
- Stephen Cole Kleene: **Introduction to metamathematics**, Bibliotheca mathematica, Wolters-Noordhoff, North-Holland, American Elsevier Pub. (1952)
- Norman M. Martin: **Direct Analogues of the Sheffer Stroke in m-valued Logic**, Notre Dame Journal of Formal Logic, vol. XVII, No. 3 (July 1976)
- Richard L. Epstein: **The Semantic Foundations of Logic Volume 1: Propositional Logics**, Kluwer Academic Publishers - Nijhoff International Philosophy series, Vol. 35 (1990)
- Newton C. A. Da Costa, Jean-Yves Béziau, Otávio A. S. Bueno: **Malinowski and Suszko on Many-valued Logics: On the Reduction of Many-valuedness to Two-Valuedness** Modern Logic, Vol. 6, no. 3 (July 1996)
- Donald E. Knuth: **The Art of Computer Programming, Volume 4, Fascicle 0B**, Addison-Wesley 2008
- João Marcos: **What is a Non-truth-functional Logic?**, Studia Logica (2009) 92, pp. 215-240 (Springer)
- Graham Priest: **An Introduction to Non-Classical Logic**, Cambridge University Press (2001)
- Lloyd Humberstone: **The Connectives**, The MIT Press (July 29, 2011)
- John Slaney on automated reasoning